

USBprog 5.0

Open-Source-Programmer mit Webinterface

Benedikt Sauter <sauter@embedded-projects.net>

Jens Nickel <jens.nickel@eimworld.com>

Der Open-Source-Programmer USBprog erfreute sich bei Entwicklern großer Beliebtheit, denn er war als „Multi-Tool“ für die unterschiedlichsten Controller zu gebrauchen. Hier kommt eine neue Version, die sogar mit einem Webserver ausgestattet ist. Dank der HTML-Oberfläche arbeitet der Programmer mit allen PC-Betriebssystemen und sogar Tablets zusammen, auch die Installation eigener Software entfällt. Eine Schnittstelle zur Automatisierung, eine Speichermöglichkeit für Hex-Files und die Integration des GDB-Debuggers für ARM sind weitere interessante Features.

Einleitung

Immer mehr Geräte werden mit Webservern ausgestattet. Über die ausgelieferten HTML-Seiten lassen sie sich dann von einem Computer aus konfigurieren und steuern. Auf dem Rechner muss dazu keine Software installiert werden, ein Standardbrowser genügt. Das Ganze funktioniert mit jedem Betriebssystem und auch vom Tablet aus. Diese schöne Idee hat die

Eigenschaften

- AVR-Programmer (avrdude [5])
- ARM-JTAG-Debugger und -Programmer (openocd [4])
- Updatefähigkeit: Weitere unterstützte Controller in Vorbereitung
- Pegelwandler (einstellbar 1,8 V, 3,3 V und 5,0 V)
- Browseroberfläche für einfache Bedienung
- Automatische Bedienung per Kommandozeilen-Tool
- Verwendung aus Atmel Studio und anderen Programmen heraus
- Speichermöglichkeit für häufig verwendete Hexfiles

Embedded Projects GmbH nun auf die Welt der Programmer ausgedehnt. Das Flashen und Auslesen von Firmware, das Setzen und Lesen von Fusebits und dergleichen mehr, das alles kann beim USBprog 5.0 über eine einfache HTML-Benutzeroberfläche erfolgen. Selbstverständlich hat das Team um Benedikt Sauter auch bei der neuesten Version des bekannten USBprog wieder voll auf den Open-Source-Gedanken gesetzt.

USBprog 5.0
OpenOCD
(Bildquelle:
Elektor)



Anzeige



Frontplatten in Profiqualität

Ab einem Stück und zu einem fairen Preis!
Einfach unseren kostenlosen Frontplatten
Designer auf www.schaeffer-ag.de
herunterladen, Frontplatte entwerfen
und direkt bestellen.



Gratis
Frontplatten
Designer

SIE DESIGNEN – WIR FERTIGEN

www.schaeffer-ag.de

Anschluss am PC

Der USBprog 5.0 ist nichts anderes als ein kleines Linux-Board im formschönen Gehäuse. Der kleine Kasten wird über USB mit dem PC verbunden und hierüber auch mit Strom versorgt. Per USB wird eine Netzwerkschnittstelle emuliert; unter Mac und Linux wird das Gerät automatisch gefunden und verbunden. Auf dem USBprog läuft ein DHCP-Server, der dem PC eine IP-Adresse zuteilt. Danach kann über die Adresse 10.0.0.1 per Browser auf den Programmierer zugegriffen werden.

Unter Windows springt beim ersten Anschließen der bekannte Treiber-Dialog auf. Hier gibt man an, dass man einen Treiber manuell auswählen möchte. Als Geräteart wählt man „Network adapters“ bzw. „Netzwerkarten“. Anschließend erscheint eine Auswahl von verschiedenen Herstellern. Hier wählt man „Microsoft Corporation“ und „Remote NDIS Compatible Device“.

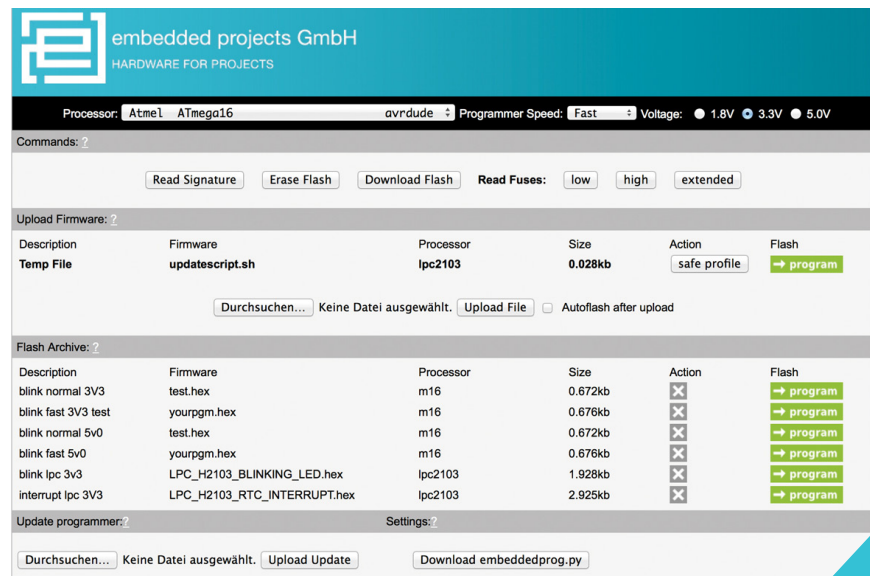


Abb. 1: Der Programmierer lässt sich über eine HTML-Oberfläche steuern, die in jedem Browser dargestellt werden kann. (Bildquelle: Elektor)

Für ARM und AVR

Nachdem die Netzwerkverbindung steht, kann man sich über einen Browser (<http://10.0.0.1>) mit dem Programmierer verbinden und sieht eine Oberfläche wie in Bild 1. Anschließend verbindet man den USBprog mit dem Zielprozessor; das Target wird dann auch mit Strom versorgt. Aktuell werden ARM-Prozessoren und Atmel-Mikrocontroller unterstützt; später werden noch PIC-Controller hinzukommen, ein Firmware-Update ist bereits in Vorbereitung.

Für die AVR-Controller bringt der Programmierer den üblichen 10-poligen ISP-Wannenstecker mit (Bild 2), über ein Flachkabel geht es weiter zum entsprechenden Gegenstück auf dem Zielboard. Um sich auch mit 6-poligen ISP-Steckern verbinden zu können, bringt der USBProg eine kleine Adapterplatine mit. Die Wannenstecker muss man sich selbst einlöten, was aber keine großen Lötkünste erfordert.

Zum Flashen und Debuggen von ARM-Controllern ist ein Adapter auf einen 20-Pin-JTAG-Steckverbinder erforderlich, der ebenfalls zum Selbst-Löten mitgeliefert wird.

Daneben ist der updatefähige Programmierer bereits mit einem 14-poligen GnuBlin/EEC-Steckverbinder ausgestattet, über den sich später einmal die bekannten GnuBlin-Erweiterungsboards steuern lassen werden – natürlich ebenfalls per Weboberfläche.

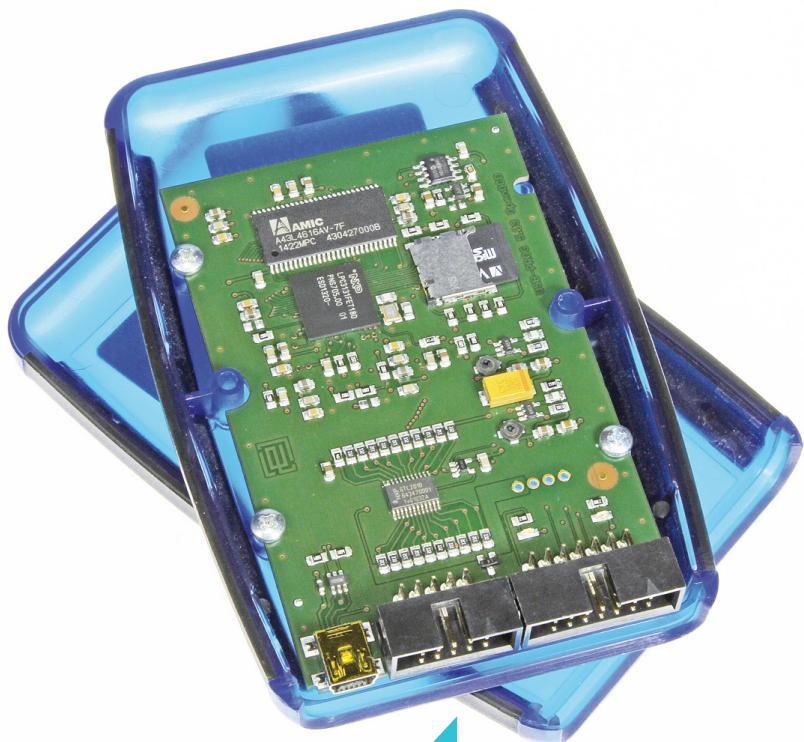


Abb. 2: Anschlüsse des USBprog 5.0. Rechts unten erkennt man den 14-poligen GnuBlin/EEC-Steckverbinder; nach einem Firmware-Update werden sich später auch einmal die GnuBlin-Erweiterungsboards ansteuern lassen. (Bildquelle: Elektor)

Manuelle Bedienung

In der Browseroberfläche wählt man nun zuerst den Prozessor aus, hierfür ist die Auswahlbox ganz oben zuständig. Um beispielsweise einen ATmega328P (Arduino Uno [1]) oder ATmega328 (T-Board 28 [2]) auszuwählen, reichen jeweils ein paar Buchstaben. Mit

den Radio-Buttons rechts daneben wird die Spannung des Zielprozessors ausgewählt.

Zuerst sollte man probeweise einmal die Signatur des Controllers oder die Fuses auslesen. Es sollte sich nach ein paar Sekunden jeweils ein schwarzes Textfenster mit dem Ergebnis öffnen (Bild 3). Danach geht es an das Programmieren eines ersten Hex-Files, hierfür ist der Abschnitt „Upload Firmware“ der Benutzeroberfläche zuständig. Mit „Browse“ wählt man ein File aus, mit „Upload File“ wird dieses zuerst auf den Programmer gespielt und schließlich mit dem grünen Button „-> program“ auf den Zielprozessor übertragen. Man kann das Hex-File auch zusammen mit einem kurzen Titel im Programmer abspeichern, wenn man auf den Button „safe profile“ klickt. Die abgespeicherten Files erscheinen im Abschnitt „Flash Archive“. So lassen sich zum Beispiel mehrere Kleinserien von Produkten recht rasch mit der jeweiligen Firmware versorgen.

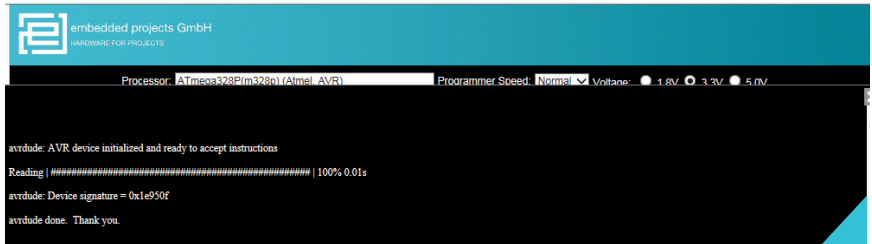


Abb. 3: Wie man hier sieht, nutzt der Programmer das Open-Source-Programmierinterface avrdude. (Bildquelle: Elektor)

Bedienung per Skript

Die manuelle Bedienung des Programmers im Webbrowser ist komfortabel, deckt aber noch nicht alle Anwendungsszenarien ab. Beim Entwickeln von Firmware arbeitet man meist mit einer Umgebung wie Atmel Studio; da spart es Zeit, wenn man aus der IDE heraus das kompilierte Programm auch gleich in den Controller flashen kann. Auch das automatisierte Flashen per Batch-Datei oder eigener PC-Software sollte mit dem Programmiergerät natürlich möglich sein.

Daher hat das Team um Benedikt Sauter ein kleines Tool in Python geschrieben, das auf dem Entwicklungsrechner per Kommandozeile aufgerufen wird. Ein Vorteil von Python ist, dass es Interpreter für alle gängigen Betriebssysteme gibt. Auf einem Mac oder Linux-Rechner kann man „embeddedprog.py“ direkt ohne zusätzliche Installation nutzen. Dort gehört Python zu den Standards des Betriebssystems. Man muss lediglich das Python-Tool selbst auf den Rechner spielen, was man über die Browseroberfläche mit dem Button „Download embeddedprog.py“ erledigen kann.

Auf der Kommandozeile ruft man nun den Python-Interpreter auf, der das Python-Programm ausführt. Der Prozessor

und die Aktion, die der Programmer ausführen soll, werden als Parameter mitgegeben. Um beispielsweise die Signatur eines ATmega328P auszulesen, lautet der Aufruf (aus dem Ordner heraus, wo das Tool abgelegt ist):

```
python embeddedprog.py \
--processor m328p --speed 2
```

Weitere Kommando-Parameter sind auf der Web-Oberfläche unten aufgeführt.

Wichtig: Beim ersten Aufruf des Python-Tools muss einmalig die IP-Adresse des Programmers angegeben werden. Diese wird anschließend lokal in einer Konfigurationsdatei gespeichert und ab da automatisch mit angefügt.

```
python embeddedprog.py \
--eeprog-ip 10.0.0.1 \
--eeprog-port 8888 \
--processor m328p --speed 2
```

Unter Windows muss zuerst Python installiert werden [3]. Das Python-Tool embeddedprog.py kann man sich herunterladen und an passender Stelle ablegen, auf der Kommandozeile muss dann

aber der Pfad auf den Python-Interpreter und auf das Python-Tool mitangegeben werden. Komfortabler fährt man mit einer Exe-Datei, die auf dem Windows-Rechner installiert wird (die passende Setup-Datei lädt man sich über den Button „Download setup.exe“ herunter). Bei der Installation werden Windows auch die Pfade bekannt gemacht; außerdem wird die Konfigurationsdatei mit der IP-Adresse angelegt. Auf der Windows-Kommandozeile (die man im Startmenü mit „cmd“ erreicht) ruft man nun einfach die Exe-Datei auf:

```
embeddedprog.exe \
--processor m328p --speed 2
```

Mit vielen Programmiersprachen lassen sich Kommandozeilenaufträge absetzen (und die Ausgabezeilen auswerten), so dass man den Programmer mit selbst geschriebener Software ansteuern kann. In einer der nächsten Ausgaben stellen wir eine entsprechende Erweiterung des EFL-Konfigurators vor.

Auch die Bedienung direkt aus Atmel Studio 6 heraus ist dank des Python-Tools möglich (siehe Kasten).

Debuggen von ARM-Prozessoren

Der neue USBprog kann auch zum Debuggen von ARM-Prozessoren verwendet werden. Hierfür wird der Open-Source-Debugger openocd [4] genutzt, der auf dem Programmer läuft. Vor dem Debuggen muss man die Firmware in den Prozessor übertragen. Anschließend startet man den Debugger per Browser („Start openocd“, siehe Bild 4) oder über die Kommandozeile. Der Debugger kann entweder über ein einfaches Telnet-Interface oder über eine GDB-Schnittstelle gesteuert werden. Größere Entwicklungsumgebungen bieten hier häufig ein entsprechendes Interface. Im Internet findet man viele Anleitungen, wenn man nach „ARM Eclipse GDB“ sucht. Als „Remote Address“ muss man anstelle des Eintrags „localhost“ die IP-Adresse des Debuggers eintragen (10.0.0.1).

Das Innenleben

Bild 5 zeigt einen Blick ins Innere des Programmers: Der Hauptprozessor ist ein LPC3131, dem Arbeitsspeicher vom Typ A43L4616 (8 MB) zur Seite gestellt ist. Für die Pegelwandlung wurde der Baustein GTL2010PW ausgewählt. Er ermöglicht eine einfache bi-



Willkommen an Bord

Die Würth Elektronik Gruppe fertigt und vertreibt elektronische, elektro-mechanische Bauelemente, Leiterplatten und intelligente Systeme. Weltweit sorgen über 6.000 Mitarbeiter dafür, dass der Bereich der elektronischen und elektro-mechanischen Bauelemente mit Sitz in Waldenburg einer der erfolgreichsten der Würth-Gruppe ist.

Unter gemeinsamer Flagge!

Teamgeist? Abenteuerlust? Werden Sie Teil unserer Mannschaft!

Traineeprogramm im Bereich Elektronikvertrieb | PLZ 80-87 & 94

Traineeprogramm im Bereich Elektronikvertrieb | PLZ 90-93 & 95-96

(Junior-) Vertriebsmitarbeiter (m/w) | Großraum München

(Junior-) Vertriebsmitarbeiter (m/w) | Großraum Nürnberg

(Junior-) Vertriebsmitarbeiter (m/w) | PLZ 90-93 & 95-96

(Junior-) Vertriebsmitarbeiter (m/w) | PLZ 86-87



www.we-online.de/karriere

more than you expect

direktionale Wandlung der Signale. Um die Spannung für die Zielhardware (1,8 V, 3,3 V und 5,0 V) per Software auswählen zu können, wurde ein LDO AP2127K-ADJ verwendet, der von einem digitalen Poti MCP4131 angesteuert wird. Bis zu 300 mA Ausgangsspannung sind möglich.

Auf dem Prozessor läuft ein Embedded-Linux, auf dem wiederum Standardsoftware wie avr-dude (für das Programmieren der Atmel-Prozessoren [5]) ihre Arbeit verrichtet. Als zentrale Komponente wurde ein Server in Python geschrieben, der sich zur Auslieferung der Weboberfläche wiederum des Lighttpd-Webservers bedient. Außerdem stellt er einen Webservice zur Steuerung des Programmers bereit, mit dem wiederum die Python-Tools auf dem PC kommunizieren. Alle Quelltexte sind auf GitHub veröffentlicht [6].

Update per Weboberfläche

Um den Programmer in Zukunft einfach mit neuen Firmware-Updates versorgen zu können, wurde eine Updatefähigkeit gleich mit integriert. Hierzu lädt man sich die neueste Version des Updates (eine gepackte Datei) von der USBprog-Webseite [7] herunter. Das Update kann dann einfach über die Browseroberfläche eingespielt werden. Es wird nach dem Upload auf dem Programmieradapter entpackt und entsprechend an die richtigen Stellen im Betriebssystem kopiert.

Zusammenarbeit mit Atmel Studio

Was wäre ein Programmer für AVR-Controller, der nicht auch aus der kostenlosen Entwicklungsumgebung Atmel Studio heraus bedienbar wäre? Das klappt auch beim USBprog - dank der Ansteuerung per Kommandozeilen-Tool (siehe Text). Zwei Aktionen dürften dabei am wichtigsten sein: nämlich das Flashen von Hexfiles auf den AVR-Controller und das Aufrufen der Browser-Oberfläche von Atmel Studio aus (zum Beispiel um Settings einzugeben).

Für beide Aktionen kann man in Atmel Studio unter Tools -> External Tools einen Menüpunkt anlegen. Nach einem Klick auf den Button „Add“ ist hierfür jeweils in den Textboxen „Title“, „Command“ und „Arguments“ ein Eintrag vorzunehmen, danach klickt man auf „Apply“. Wir beginnen mit „USBprog 5.0 Program“ für das Uploaden von Firmware (siehe Screenshot). Unter „Command“ ist jeweils der gesamte Pfad auf das Tool embeddedprog.exe (siehe Text) einzutragen.

Hat man für beide Aktionen einen Eintrag wie in den Screenshots gezeigt angelegt, dann erscheinen die beiden Einträge im Atmel-Studio-Hauptmenü unter „Tools“.

Betätigt man zum ersten Mal „USBprog 5.0 Program“, dann springt die Browseroberfläche auf. Hier muss man zuerst den Prozessor und dessen Betriebsspannung einstellen. Ab jetzt kann man über diesen Menüpunkt direkt aus Atmel Studio heraus Hexfiles in den Controller flashen.

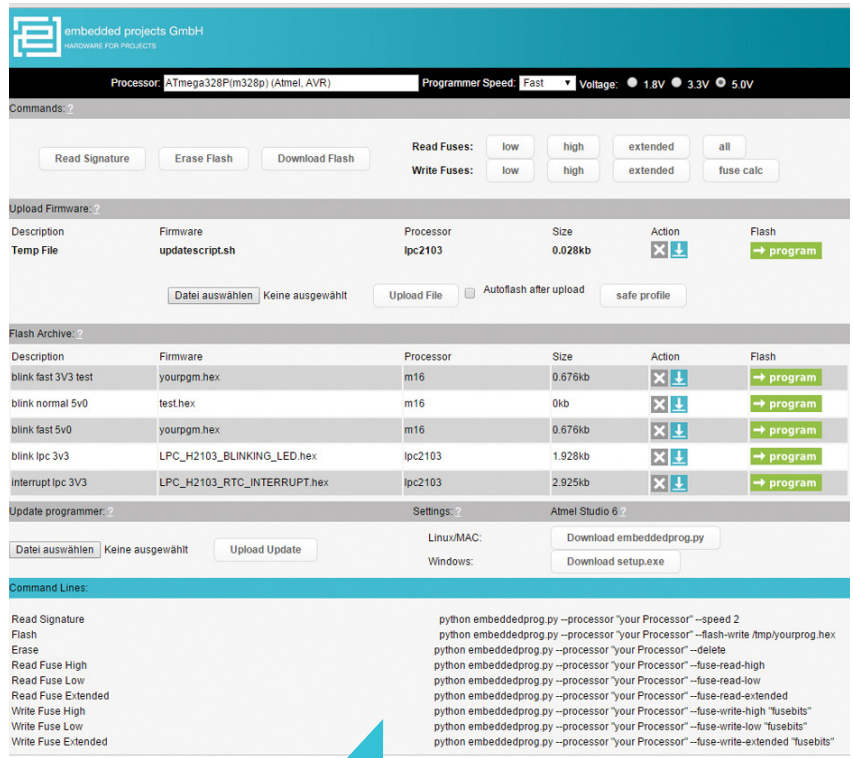


Abb. 4: Die Oberfläche nach dem Auswählen eines ARM-Controllers. (Bildquelle: Elektor)

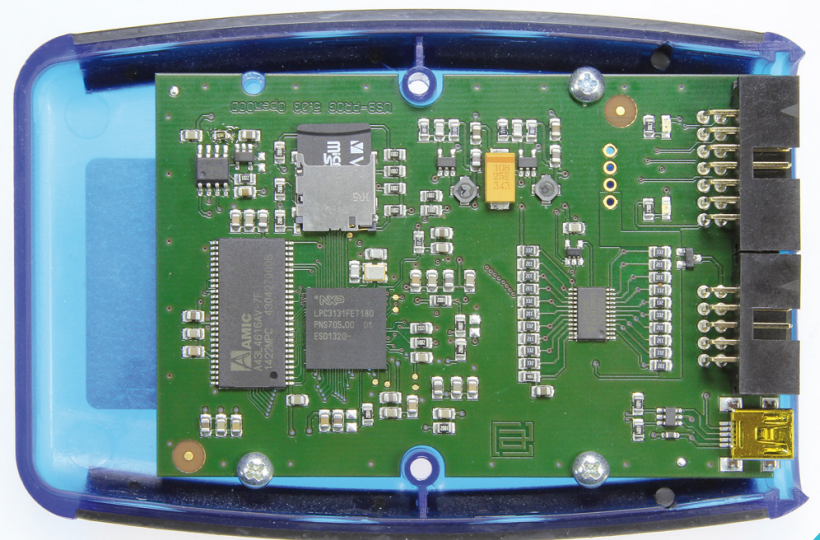


Abb. 5: Basis des Programmers ist ein LPC3131, auf dem Linux läuft. (Bildquelle: Elektor)

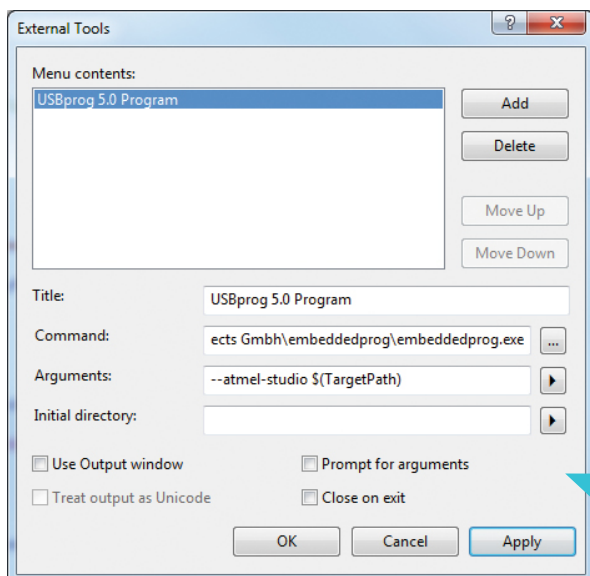


Abb. 6: In Atmel Studio ein neues „External Tool“ namens „USBprog 5.0 Program“ hinzufügen, um einen AVR aus der IDE heraus flashen zu können. (Bildquelle: Elektor)

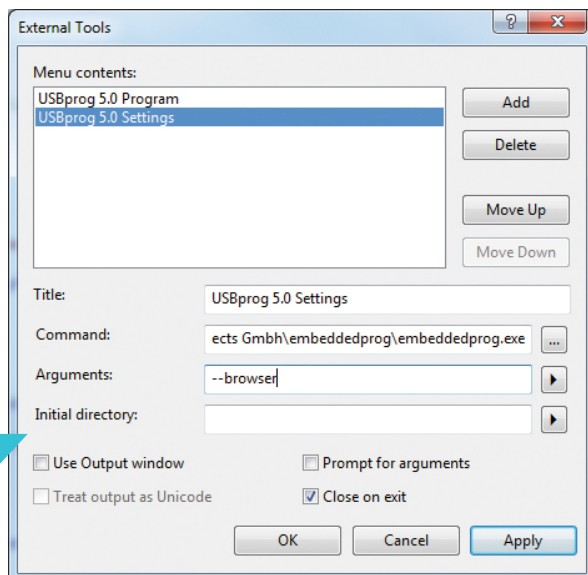


Abb. 7: Einrichten von USBprog 5.0 Settings erlaubt das Öffnen der Weboberfläche, in welcher weitere Einstellungen vorgenommen werden können. (Bildquelle: Elektor)

Weblinks

- [1] www.elektor.de/arduino-uno
- [2] www.elektor.de/t-board-28-130581-93
- [3] <https://www.python.org>
- [4] www.openocd.org
- [5] www.nongnu.org/avrdude
- [6] <https://github.com/embeddedprojects/usbprog5>
- [7] <http://usbprog5.embedded-projects.net>

Top-Studium für Embedded Systems



Kombinierte Ausbildung für Hardware & Software in Österreichs Silicon Valley, Hagenberg

Bachelor Hardware-Software-Design

Master Embedded Systems Design

- >> Top-Ranking in Österreich
- >> renommierte & moderne FH
- >> teamorientiertes Studium
- >> individuelle Talentförderung
- >> Anrechnung v. Vorkenntnissen
- >> Wohnen direkt am Campus

Schwerpunkte

- Open Source Hardware & Software •
- Microcontrollers • System-on-Chip •
- FPGA • Sensors • Actuators •
- Digital Communication • Embedded Software •
- Parallel Computing • Realtime Systems •
- System Design • Cyber-Physical Systems • Robotics



Studium mit Zukunft

www.fh-ooe.at/hsd